Bremen AG Tagung 2022 GAVO Puzzler: Solution

As usual for our puzzlers, this is a problem that's best addressed with TAP and ADQL. While you could use pyVO or Aladin or any other TAP client, we'll use TOPCAT here.

So, start TOPCAT, enter the VO/Table Access Protocol window, type "gaia dr3" into the "Keywords" field. You will see a fairly large number of services, and most of those carry the gaiadr3.gaia_source table under one name or the other. You should get the same result from all that do, but somewhat parochially, we will use the lite (as in: only the most basic columns present) version at GAVO's data centre (the "GAVO DC TAP" entry).

If you select it and hit "Use service", you will end up in a table browser; typing "gaia" in "Find" will lead you to "gaia.edr3lite" fast. Select that table in the Metadata pane. Once you have done that, you can inspect its column metadata in the columns tab, and TOPCAT will adapt its built-in examples – that's nice because the example in Basic → Cone Selection will then be written for the right table. Try it, and TOPCAT should fill the query box with

```
SELECT
 TOP 1000
 *
 FROM gaia.dr3lite
 WHERE
   1=CONTAINS(POINT('ICRS', ra, dec),
     CIRCLE('ICRS', 189.2, 62.21, 0.05 ))
```

In this example, you can simply to replace the position and the radius in the CIRCLE clause with the parameters $(246.71, -24.54, 0.25)$ from the problem sheet. Replacing the `TOP 1000 *` (for "all columns"; in general, it's a *very* good idea to *not* blindly pull all columns from a table; rather say what you actually need for your concrete problem) with `phot_bp_mean_mag-phot_rp_mean_mag` will *almost* yield the answer.

Except: It would be extremely clumsy to download the all these rows and then average them locally – the server can do this at least as well, and if it does that, you reduce the amount of data transferred by a several orders of magnitude. To work this miracle, use the aggregate function AVG. In sum:

```
SELECT
 AVG(phot_bp_mean_mag-phot_rp_mean_mag) AS avgcol
FROM gaia.dr3lite
WHERE
 1=CONTAINS(POINT('ICRS', ra, dec),
   CIRCLE('ICRS', 246.71, -24.54, 0.25 ))
```

This yields about 2.92 within perhaps a second for the cloudy field, and about 1.39 for the clear one.

In case you're not convinced that it's smarter to do the averaging on the server side, try radii of 0.5, 1, 2, and 5 degrees (the large radii may require Asynchronous mode).

Some remarks

Dealing with logarithms

Note that a simple AVG over magnitudes actually is rather unphysical; remember that the magnitude is the logarithm of the flux, and the sort of average you get when you *add* all these logarithms up

is a bit odd. It would hence be preferable to to all the adding and substracting in fluxes. ADQL makes that easy; you could write

```
SELECT
  AVG(LOG10(
    POWER(10, phot_bp_mean_mag)-POWER(10, phot_rp_mean_mag)) AS avgcol
...
```

– but take this with a grain of salt.

Finding a spot rich and poor in dust

I knew about the dusty spot from my blog post on the loneliest star in the sky[1]. For the dust-free spot, I remembered the bayestar17 dust map by Green, Schlafly et al[2]. This still turned out to be a bit tricky and had results suggesting certain systematics in the dust map. This was (well: to me) interesting enough to merit another blog post[3].

Doing it in one go

TOPCAT's author has submitted a solution that covers both fields in one go. While it looks like a bad hack, this pattern actually is useful in several other circumstances. So, have a look at

```
select bp_rp_monoceros, bp_rp_bootes
from (
      select avg(phot_bp_mean_mag - phot_rp_mean_mag) as bp_rp_bootes
      from gaia.dr3lite
      where distance(214.85, 33.68, ra, dec) < 0.25
) as bs
join (
      select avg(phot_bp_mean_mag - phot_rp_mean_mag) as bp_rp_monoceros
      from gaia.dr3lite
      where distance(246.71, -24.54, ra, dec) < 0.25
) as ms
on 1=1
```

Admittedly, this does not scale well to hundreds of positions. For that, see the next section, even if it is just a pointer.

Drawing colour maps

Did you like the map of average colour indexes we printed on the problem sheet? We'll show how to produce these within the next few weeks on our blog. So, be sure to subscribe to our RSS feed[4].

[1] https://blog.g-vo.org/the-loneliest-star-in-the-sky.html
[2] cf. https://blog.g-vo.org/deredden-using-tap.html
[3] https://blog.g-vo.org/find-a-dust-free-window-using-adql.html
[4] https://blog.g-vo.org/feeds/all.atom.xml